Numerical results:

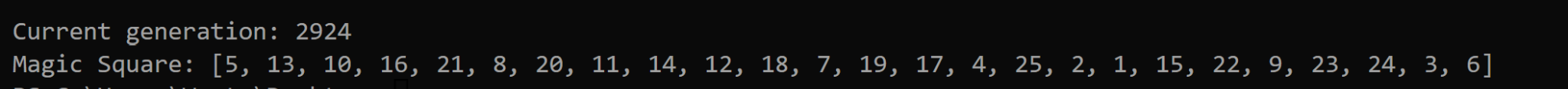
Our magic squares were represented internally as a 1d array

5x5:

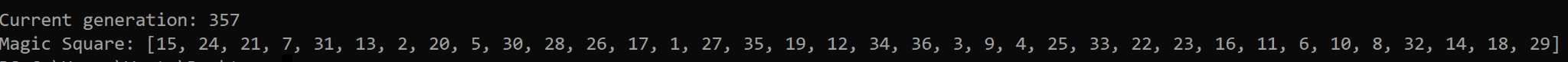
We ran this multiple times and the generations it took:

2924

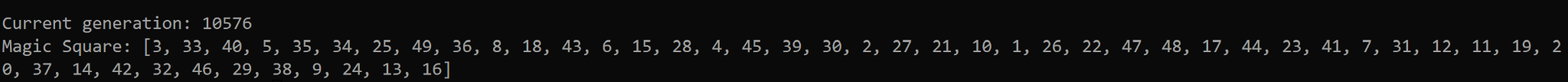
2342



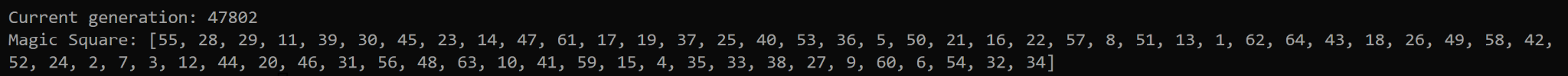
6x6:



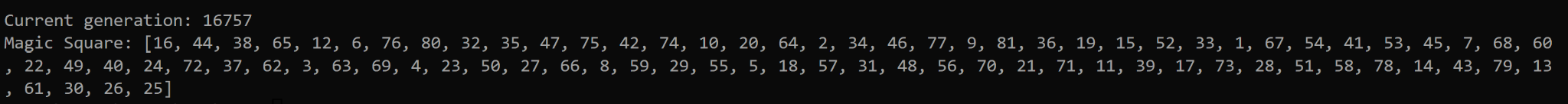
7x7:



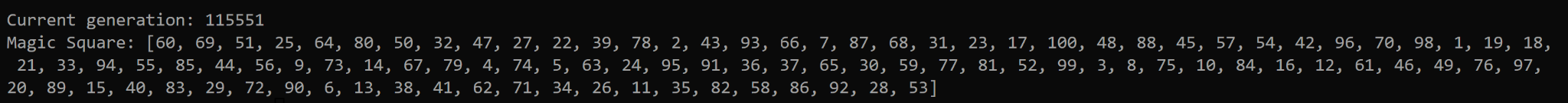
8x8:



9x9:



10x10:



The description of the genetic operators used:

Population size of 200

We introduced elites into the selection process of the next generation. We found that a very high number of elites, about 75% of the population worked well.

Due to the high number of elites(exploitation), in order to keep genetic diversity, NO duplicate squares are allowed in the next generation. In addition to this, the mutation chance is a 50% chance to increase our exploration. Our implementation originally allowed duplicate squares but quickly the population would converge on a local optimum.

The crossover function we implemented was a two-point PMX function, before our code was using a 1 point simple approach and this did not yield good results.

The mutation function randomly picks two indexes and swaps the values at those indexes.

The fitness function we used makes use of the fact that we can know the magic number of a square based on the size of the dimensions. So we calculate this magic number and determine the distance away for the magic number of each row, column, and diagonal then we sum these distances giving us our fitness score. This means that a fitness score of 0 is a magic square and the larger the fitness value, the worse the square is.

The selection function used to pick the parents for the crossover is based on the fitness proportional selection. Where everyone in the population has a chance to be selected, but the better the fitness of the individual, the greater the chance of being selected for the crossover function

Conclusions:

Using the implementation described, the program was able to find a magic square on the first attempt for all of the sizes shown above in the pictures. It is the large amount of elites that gives us our exploitation, in addition to the fitness proportional selection for our crossover function. Because of the high number of elites, the exploitation of this program is very high. To combat this, no duplicate squares are allowed into the next generation to increase our exploration. And a more sophisticated crossover function is used(PMX). However, with just these two exploration tools used it is still not enough to introduce more genetic diversity which is why the mutation chance is very high (50%). Altogether this provided a very efficient solution for finding magic squares as opposed to our previous implementation that allowed duplicate squares, had a low mutation rate(1/100), and a small percentage of elites (5%) for similar population sizes.